Weekly Report(April.18,2019-April.30,2019)

Rui Shaopu

Abstract

After finishing three assignments, I found I'm a little confused about some models, therefore I reviewed the videos and write this summary report to summarize cs231n.

1 models

The following are some important sentences I think. So I write them down for later review.

1.1 Linear Classification

Linear Classification has this interpretation as learning templates per class. For every pixel in the image, and for every one of our 10 classes, there exists some entry in this matrix W,telling us how much does that pixel influence that class. So that means that each of these rows in the matrix W ends up corresponding to a template for the class. And if we take those rows and unravel, so each of those rows again corresponds to a weighting between the values of, between the pixel values of the image and that class, so if we take that row and unravel it back into an image, then we can visualize the learned template for each of these classes.

We also has this interpretation of linear classification as learning linear decision boundaries between pixels in some high dimensional space where the dimensions of the space correspond to the values of the pixel intensity values of the image.

1.2 NN

This kind of multiple layer network lets you do is each of this intermediate variable h, W1 can still be this kinds of templates, but now you have all of this scores for these tamplates in h,and we can have another layer on top that's combining these together

1.3 CNN

how to choose stride? At one sense it's kind of the resolution at which you slide it on and usually the reason behind this is because when we have a larger stride what we end up getting as the output is a down sampled image, and so what this downsampled image lets us have is both, it's kind of like pooling in a sense but it's a different and sometimes works better way of doing pooling is one of the ituitions behind this,

1.4 Activation Function

Sigmod: 1.Staurated neurons "kill" the gradients 2.Sigmod outputs are not zero-centered 3.a little bit computationally expensive

tanh:still kills gradients when saturated.

ReLU: not zero-centered output An annoyance some neurons will die since their gradients are 0 caused by ReLU

1.5 Batch Normalization

reason: if not, the loss function is extremely sensitive to small perturbations in linear classifier in our weight matrix. After Batch Normalization, our loss function will be less sensitive to small perturbations in the parameter valus.

1.6 Optimization

1.6.1 Problems with SGD

In fact, you get very slow progress along the horizontal dimension, whichis the less sensitive dimension, and you get this zigzagging, nasty, nasty zigzagging behavior across the fast-changing dimension. This problem becomes much more common in high dimensions. local minima and saddle point are also its problems. SGD will get stuck because at this local minima, the gradient is zero because it's locally flat.

1.6.2 SGD + Momentum

When functions are at saddle point or local minima, although this point will not have positive gradient, it will still have velocity, so we can get over this local minima or saddle point and continue forward.

1.7 Regulazation

L2 regulazation: $R(W) = \sum_k \sum_l W_{k,l}^2$ (Weight decay)

L1 regulazation: $R(W) = \sum_k \sum_l |W_{k,l}|$

Elastic net(L1 + l2): $R(W) = \sum_k \sum_l \beta W_{k,l}^2 + |W_{k,l}|$

1.7.1 Dropout

In each forward pass, randomly set some neurons to zero. Probability of dropping is a hyperparameter; 0.5 is common.

The network cannot depend too much on any of these one features. instead, it kind of needs to distribute its idea of catness across many different features. This might help prevent overfitting somehow.

Dropout is trainning a large ensemble of models(that share parameters).

1.7.2 Transfer Learning

One problem with overffing is sometimes you overfit because yoou don't have enough data. You want to use a big, powerfull model, but that big, powerfull model just is going to overfit too much on your small dataset.

1.8 CNN Architectures

When we take these small filters now we have fewer parameters and more filters and we try and stack more of them instead of having larger filters.

Stack of three 3x3 conv(stride 1) =layers has same effective receptive field as one 7x7 conv layer.

1.9 RNN

one to many model: input is some object of fixed size, like a image, but output is a sequence of variable length, such as a caption.
many to one model: input is variably sized, this might be something like a piece of text, or a video

which might have a variable number of frames.
many to many: both input and output are variable in length.we might see in machine translation.

Rather than just doing a single feed forward pass and making a decision at once, network is actually looking around the image and taking various glimpses of different parts of the image. After making some series of glimpses, then it makes its final decision.

1.10 DeepDream

Actually There is nothing to record in this part. However, after learning this part again, I understand what different layers of neural network have learned during the training process. The higher layer, the more features it can learn. DeepDream algorithm makes it visible.

2 Summary

The learning process for this summary really makes me understand more. I know more details which I didn't pay attention to. After this summary, I think my training process could come to an end and I have prepared to learn about mini-project.